Analysis of digital chaotic optical signals

J.A. Martin-Pereda*, A. Gonzalez-Marcos
E.T.S. Ingenieros de Telecomunicación. Universidad Politécnica de Madrid

ABSTRACT

The main objective of this paper is to present some tools to analyze a digital chaotic signal. We have proposed some of them previously, as a new type of phase diagrams with binary signals converted to hexadecimal. Moreover, the main emphasis will be given in this paper to an analysis of the chaotic signal based on the Lempel and Ziv method. This technique has been employed partly by us to a very short stream of data. In this paper we will extend this method to long trains of data (larger than 2000 bit units). The main characteristics of the chaotic signal are obtained with this method being possible to present numerical values to indicate the properties of the chaos.

Keywords: Digital chaos, Chaotic signals, Lempel and Ziv method

1. INTRODUCTION

The analysis of chaotic signals has been always performed by analytical methods almost since the beginning of their study. This was due to the analogue character of the involved signals. But in some cases, the chaotic signal has a digital characte and the conventional methods are no longer valid. This is the case for the chaotic signal obtained from Optical Logic Cells when some feedback is applied from one of the possible outputs to one of the control gates. We have studied this situation in several papers. The main problem with this signal is that analytical methods cannot be applied to the resulting chaotic signal. To extract its characteristics is not possible with the methods employed in other analogue cases.

We have proposed several methods in the last years[1-2,4-5]. Every one of them has been based in a new type of phase diagram where binary data was converted to a hexadecimal system. The binary signals are, in this way, converted to a multilevel signal. Signals at a particular time are represented as a function of the signal at the previous time interval. A periodic signal appears, in this representation, as a closed figure and a chaotic one offers a time evolution without an particular recovery of previous positions. This type of analysis is convenient when just an indication about the signal characteristics need to be known.

But in some occasions quantitative data may be needed. The above indicated method can not give this type of information. Hence, it is necessary to adopt some other strategy to get the corresponding results. This is the aim of this paper. In order to do that, we have adopted a similar method to the one employed previously by us[1]. It is based on the Lempel and Ziv Complexity Measure and at our previous paper it was applied to a short string of data. In this case, a much longer string has been taken and the numerical method employed may be applied to other similar situations.

Our paper will be divided into two different parts. In the first one, some ideas about the Lempel and Ziv method will be given and how it can be applied to the measure of chaotic signals. The second one wil l deal with its application to the digital chaos obtained previously by us from an Optical Programmable Logic Cell.

2. THE LEMPEL AND ZIV COMPLEXITY MEASURE

The first approaches to study the complexit of a signal were based on the Kolmogorov and Chaitin orks. Most of them performed an association of the signal complexity with the complexity of an algorithm able to generate a similar string. In some other cases, the association was with the binary code performing that generation. The appearance of the Lemp el and Ziv algorithm[3] (LZ from now on) drastically changed this type of approach.

The change in the measure method introduced by the LZ algorithm was to associate the string complexity with the number of needed substrings to generate the initial string. Moreover, it is related too with the number of different substrings

* Correspondence: Email jamp@tfo.upm.es; E.T.S. Ingenieros de Telecomunicación. Universidad Politécnica de Madrid, Ciudad Universitaria. 28040 Madrid. Spain.

Mathematics of Data/Image Coding, Compression, and Encryption IV, with Applications,
Mark S. Schmalz, Editor, Proceedings of SPIE Vol. 4475 (2001) © 2001 SPIE · 0277-786X/01/$15.00

that have appeared and its apparition rate. Due to the characteristics of the LZ method, some words about Information Theory are needed, at least to remember them and to clarify the posterior notation.

2.1. Some concepts from Information Theory

Be A^* the set of any finite string generated with an alphabet A. In our present case, this alphabet is composed by just two symbols, namely "0" and "1".

Define:

- $A^n = \{S \in A^* \mid l(s)=s\}$ being "l" the string length
- $S = s_1 s_2 s_3 ... s_n$ where $S \in A^n$ y $s_i \in A$. S is the string and s_i is each one of the element of that string.
- $S = s_i s_{i+1} ... s_j$ is the substring going from the i-th element in S to the j-th.
- $S(i,j) = \wedge$ is the null substring if $i > j$
- $S = QR = q_1 q_2 ... q_m r_1 r_2 .... _n = s_1 s_2 ... s_n$ S is the result to concatenate two strings, Q and R, and verifies that Q = S(1,m) and R = S(m+1, m+n)
- As a consequence, it is possible to define $S^2 = ss$ and $S^0 = \wedge$
- We say that a string Q is prefix of other string $S \in A^*$ and that S is an extension of Q if there is an index "i" such that Q = S(1,i). If l(Q) > l(S), Q and S are prefix and proper extension.
  $S\pi^i = S(1, l(s) - i)$ is the result to eliminate the last i symbols from the string S.
  Hence we define
  $s\pi^o = S$ and $s\pi^i = \wedge$ if $i \geq l(S)$
- The vocabulary of a string is the set of substrings from S with the form S(i,j)
- Characteristic words from the strings of a vocabulary are the words no belonging to any proper prefix of S. The set of these characteristics words is the characteristic vocabulary.

We propose an example to clarify the above definitions.

Be 0010 the string to be studied. The substrings will be:

$S = \{0010\} \rightarrow$ string

$\vee(0010) = \{\wedge, 0,1,01,10,00,001,010,0010\} \rightarrow$ vocabulary

$\{0,00,001\} \rightarrow$ Proper prefix for S

$\vee(0) = \{\wedge,0\}$		Prefixes for the corresponding vocabularies
$\vee(00) = \{\wedge,0,00\}$	}	
$\vee(001) = \{\wedge,0,1,00,01,001\}$		

$e(0010) = \{10,010,0010\} = \vee(S) - \vee(S\pi)$

It is possible to verify from above facts that

$$\vee(S\pi) \subset \vee(S)$$

and hence

$$e(S) = \vee(S) - \vee(S\pi)$$

Define now

Reproduction: Given a string S and an extension of that string R (R = SQ) such that q belongs to the SQπ vocabulary, then is reproducible from S and is designed as S $\rightarrow$ R. In this case there is an index "i" that verifies

Q = R (I, l(Q)+I-1)

For instance: with I = 2 : 001 $\rightarrow$ 00101010.

Production: A no null string is producible from one of the prefixes, if S(1,j) $\rightarrow$ Sπ and j < l(S). It will be written S(1,j) $\Rightarrow$ S. S(1,j) is named base of S.

An example may clarify these points. Be 01 $\Rightarrow$ 0100.

S(1,j) = 01
Sπ = 010
S(1,j) $\rightarrow$ Sπ with i = 1.

Next step is to look for a way to reconstruct the string. A possible method is to find a history of the string

$H(S) = S(1,h_1)S(h_1+1,h_i)$ with $_1 = 1$ and $h_m(l(S)$

Each string is named "*component*". If we find components $Hi(S)=S(h_{i-1},h_i)$ such that they verif

$S(1,h_{i-1}) \Rightarrow S(1,h_i)$ but $S(1,h_{i-1}) \not\rightarrow S(1,h_i)$, where $\not\rightarrow$ is the negation of $\rightarrow$, these components are named "*exhaustive*" of the string. The number of this strings gives the signal complexity.

As an example of previous concepts we present the exhaustive history of the string

S = 0001101001000101

Q	SQ	Q∈−∉∨π)	S
-	-		{0}
0	00	∈	{0.001}
00	000	∈	
001	0001	∉	
1	00011	∈	{0.001.10}
10	000110	∉	
1	0001101	∈	{0.001.10.100}
10	00011010	∈	
100	000110100	∉	
1	0001101001	∈	{0.001.10.100.1000}
10	00011010010	∈	
100	000110100100	∈	
1000	0001101001000	∉	
1	00011010010001	∈	{0.001.10.100.1000.101
10	000110100100010	∈	
101	0001101001000101	∉	

In this case, the resulting complexity is 6. According to LZ the minimum value for complexity is c(S) = 2.

But this number has not a real sense. Only relative values of c(S) are meaningful and in particular it is the comparison with the c(S) for a random string that is meaningful. Lempel and Ziv have shown that for a random string of length l, the LZ complexity is given b

$$b(l) = \frac{hl}{\log_K(l)}$$

where K denotes the number of elements in the alphabet and h denotes the normalized source of entropy. In ou case, the number of elements in our alphabet is 2, namely, "1" and "0". The normalized source of entropy is the general definition of information, given in terms of the probabilities of the various states of the system (Shannon and Weaver, 1949), divided by the maximum information obtained when each state is equally probable. Assuming that p_i denotes the probability that the system is in the ith state, then

$$I = -\sum_i p_i \lg p_i$$

Hence

$$h = \frac{-1}{\lg N}\sum_{i=1}^{N} p_i \lg p_i \leq 1$$

So, the normalized entropy h is determined by determining the probability p_i for each state i. Counting the occurrences of each symbol in the alphabet and then dividing by the total number of symbols in the string obtain this probabilit . In the case that each symbol from the alphabet is equally probable, then p = 1/N and h = 1. Comparing with the complexity for a random string, we have to compute

$$\lim_{n\to\infty}\left[\frac{c(S)}{b(N)}\right]$$

for a string with n elements. If the ratio is less than 1, then we can conclude that this is due to a pattern formation in the string S.

3. SOURCE OF DIGITAL CHAOS

We have previously reported the source studied with above indicated method in several places. It is composed by two non linear devices – an "on-off" and a "SEED-like" – arranged in a structure able to offer two logical outputs from two binary inputs. They will be employed in our present work. Its basic configuration appears in Fig. 1. Input and output signals are optical binary data and the employed devices, non-linea optoelectronic devices with digital characteristics. Fig. 1 shows a block representation of the basic cell. Two optical devices, P and Q, with a non - linear behaviour, compose the circuit. The output of each one of them correspond to the two final outputs, $_1$ and O_2, of the cell. The possible input to the circuit are four. Two of them are for input data, I_1 e I_2, and the other two, g and h, for contro signals. The corresponding inputs to the non-linear devices, P and Q, are functions of these signals plus, in the case of the P device, one other coming from inside the own cell and obtained from the Q device. Some details about the internal connections of th cell are shown in Fig. 2. More details may be seen in [2-3]. Two more gates are added as control inputs. This configuration, when arranged in a conventional way, gives fifteen pairs of possible

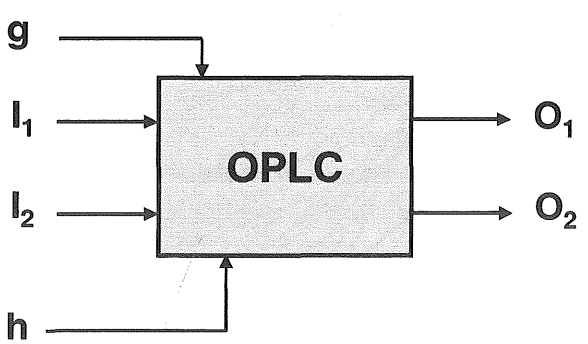

Fig. 1.- General configuration of the Optically Programmable Logic Cell.

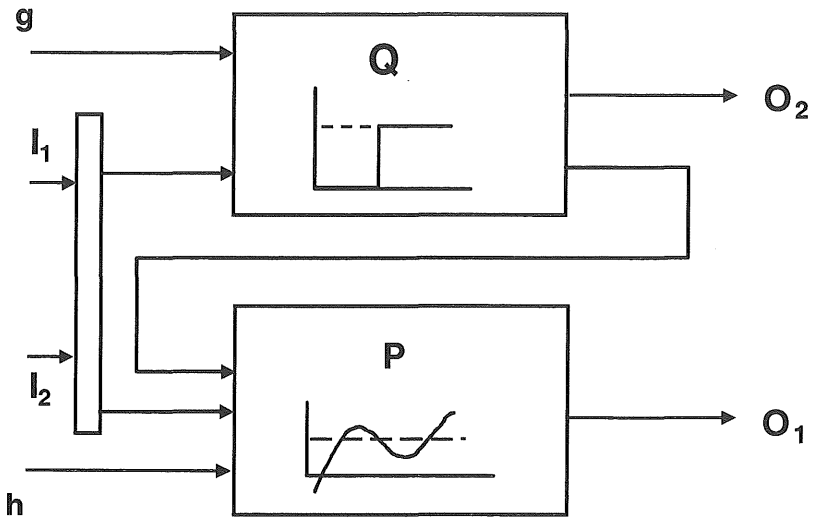

Fig. 2.- Internal configuration of the OPLC.

outputs, with Boolean functions of the input gates. This situation is drastically changed when a feedback is added betwee one of the possible outputs and one of the control gates. Depending on the external delay time and its relation with the internal time of the structure, different results are obtained. If a multilevel input is added to the input and when the intern al time is much shorter than the external one, a very irregular output appears.

The problem to characterize this signal comes from its binary form. Solutions adopted when the output has analog form are well known. If its characteristics are chaotic, several methods have been proposed to analyze it. Techniques as the phase diagram may be found at any Chaos Textbook. Moreover, numerical characterization is possible, for instance, with

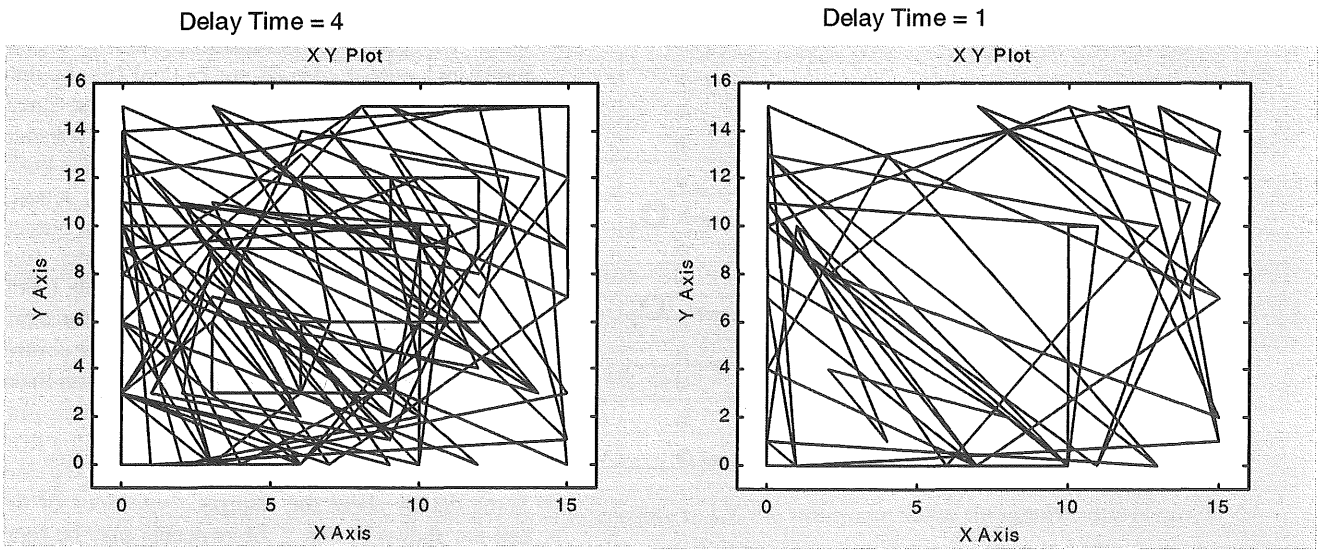

Figure 3.- Phase Diagram for different delay times.

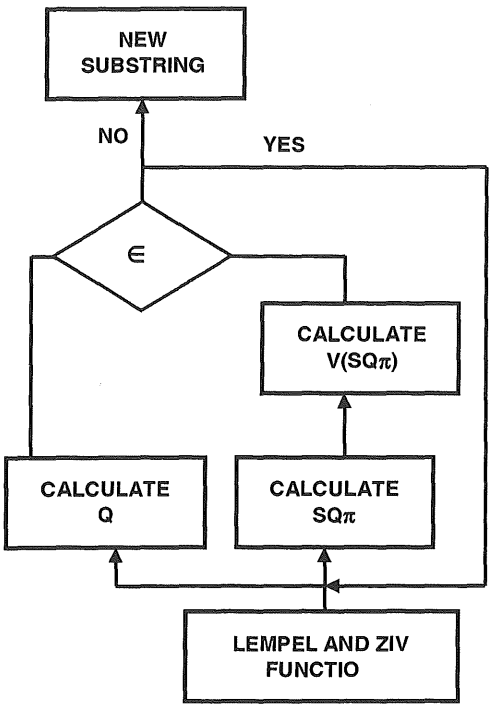

Fig. 4.- Flow chart corresponding to the Lempel and Ziv method.

the Lyapunov Exponent.

But the situation is very different when the output has a binary character and it is intended to know if it is a chaotic signal. We have solved this problem by converting the binary data to an hexadecimal form. Details are given in several places. The result may now be represented in a Phase Diagram. Some of the obtained results are shown in Fig. 3 a-b. Fig. 3.a-b offe the corresponding phase diagram for two different delay times. Signals at a time interval t are represented as a function of signals at (t-1).

To extract numerical information about possible chaotic characteristics of this behavior is almost impossible. It is because that a technique as the presented before, based on the Lempel and Ziv method is useful. The final part of this paper will be this approach.

4. APPLICATION OF THE LEMPEL AND ZIV METHOD TO THE ANALYSIS OF CHAOTIC DATA

In a previous paper, a first analysis of the above indicated optically programmable logic cell was reported. But at that time, a very short string of data, namely 2048,were taken. Hence, the results were just an approximation instead real ones. As it is indicated above, the correct answer is when the number of analyzed data tends to infinity. In the present situation, the number of samples has been 11500. This represents a considerable increase in computing time if some non efficient numerical method is employed. The solution adopted by us is represented in Fig. 4. It shows the Flow Chart of the computational adopted method.

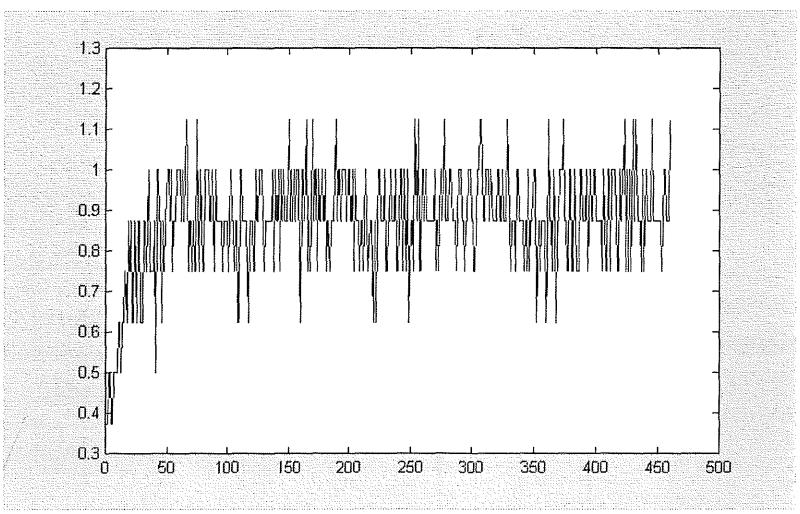

Fig. 5.- Analysis of a string with 11500 data bits grouped in packets of 25 bits.

The first step is to cut the initial total string, with 11500 data, in substrings with 25 or 50 data. This is needed in order to allow an adequate charge to the computer. The first result, after cutting the digital signal in groups of 25 appears in

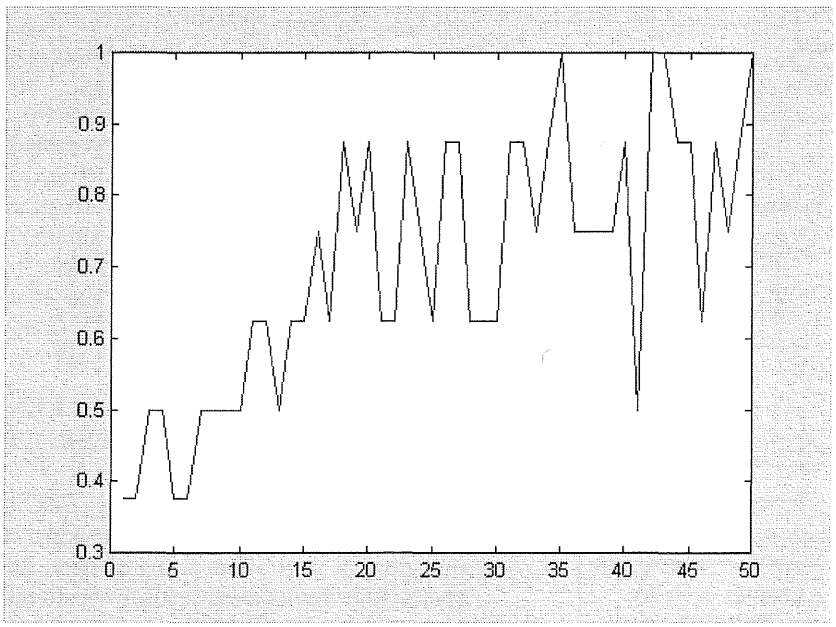

Fig. 6.- Enlarged first 50 groups of 25 bits in Fig. 5

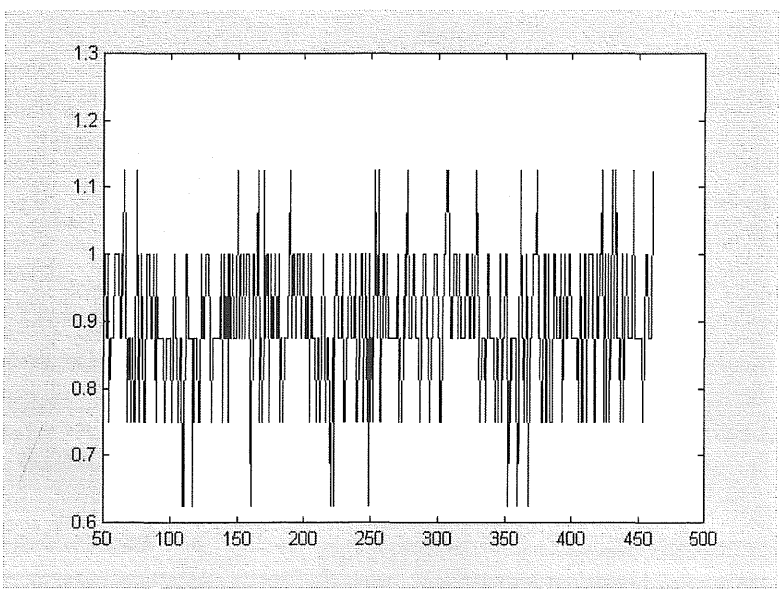

Fig. 7.- Complexity measure from the 51st group of 25 bits.

Fig. 5. It is possible to see that the result changes from a group to the following one. The number that appears in abscises is the order of the taken group. The resulting graph indicates that there is an initial time, namely the corresponding to the first 50 groups, that the obtained values go from low values fo the relative complexity to values approaching the unity. This is an indication that the chaos generator needs a certain time interval to get the final state. Fig. 6 enlarges the first part of previous figure. It can be seen that, although obtained values reach at certain sets values close to "one" other values ar much lower. On the contrary, when one consider just the obtained values for sets taken after the number 50, the results, shown in Fig. 7 offer a behavior with values higher that 0.9 in almost every case. If we calculate the mean value for those results, the obtained value is 0,8957. This implies chaos in 90 % of the total time.

Next step is to perform similar calculations with longer sets of data. In this case we have taken sets of 50 data. The number of intervals is now, as a consequence, smaller than before. The results are shown in Figs. 8-10. The general

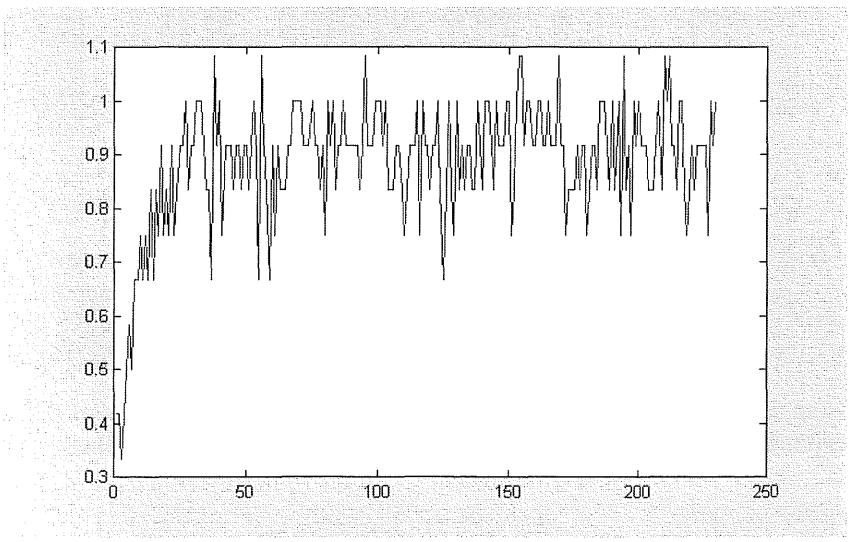

Fig. 8- Analysis of a string with 11500 data bits grouped in packets of 50 bits.

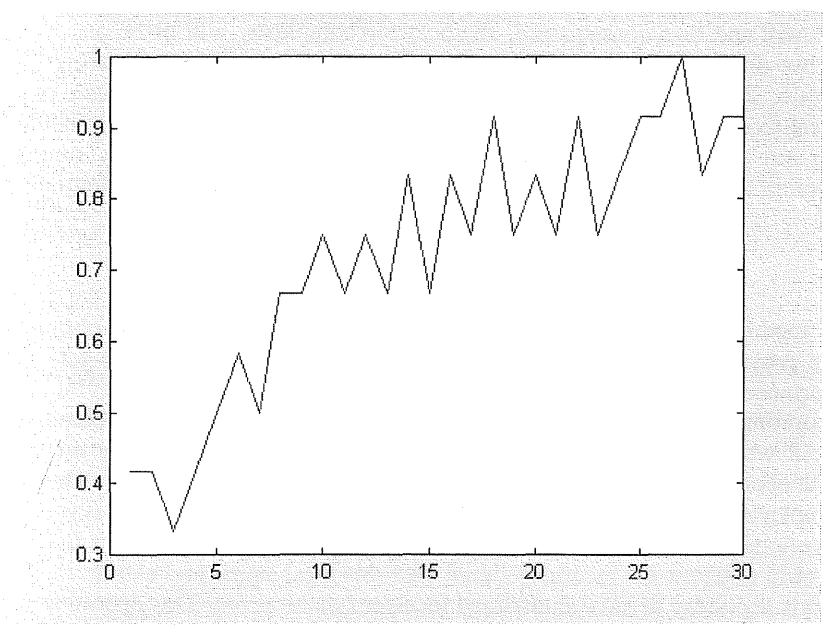

Fig. 9.- Enlarged first 25 groups of 50 bits in Fig. 8.

characteristic of the solution is similar to the previous case. But they're some significant differences. The transient state (Fig. 9) is similar to the obtained with sets of 25 data (Fig. 6). If previously we had 1250 data (25 x 50) we have now 1500

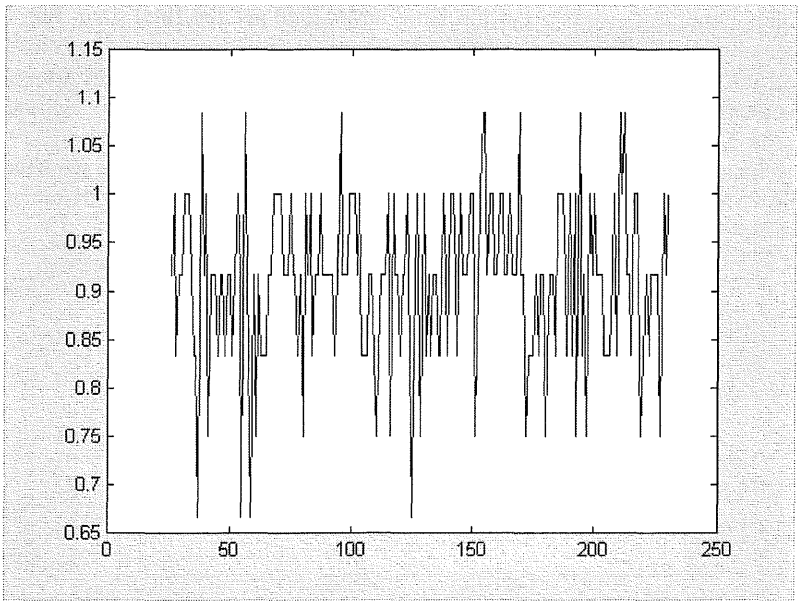

Fig. 10.- Complexity measure from the 31st group of 50 bits.

data (50 x 30). The mean value in this interval has similar value too. But the result gets a value closer to the theoretical "one" for perfect chaos if we extend it to the total number of data. The mean value is now 0.9093.

The obtained results are in a good agreement with a first intuitive interpretation. A small string of bits has to have, because the small number of possible states, a small value for the complexity. The numbers of possible words in the dictionary have to be small and, as a consequence, the obtained complexity will no be possible to be close to unity. In our case, we have studied two different approaches to the same problem. The length of the string is, in the first case, of 25 bits and 50 in the second one. It is clear, from these values that it should be difficult to reach values for the complexity very close to 1. And it is clear too, that if the number of analyzed bits is larger, the complexity should be larger, if the string has chaotic properties. It is sure that if the string should have increased to 100, the complexit number should be closer to unity. This is a clear indication of the chaotic properties of ou train of pulses as well as the potentialities of the reported method.

5. CONCLUSIONS

The reported method offers a way to quantify the characteristics of the signal obtained from a chaos generator. Moreover, it gives the possibility to analyze a portion of a long string of data and to get some information about its complexity. The importance of this method is clear. As it was pointed out at the beginning of this paper, one of the intentions when elaborating this method was to analyze the characteristics of the obtained signal from an opticall programmable logic cell when some type of feedback was present. Through some other different types of analysis we go the information that the obtained signal was, under certain boundary conditions, chaotic. This analysis was necessary when the signal was to be employed in applications as signal encryption. In these circumstances it is necessary to have an emitte able to generate a chaotic signal in order to mix it with the informat ion signal. But at the receiver it necessary to have another chaos generator to compose the incoming signal with this local signal and extract the transmitted information. To achieve this function it is necessary to have emitter and receiver synchronized. This operation is not an easy task and it has been the object of attention from several groups around the world. Two facts are important: to have synchronized both systems, emitter and receiver, and to maintain the same chaos at them. We have proposed in a recent work[6] a method to achieve this goal by controlling the resulting chaos by a Lempel and Ziv complexity measure system. The possibility to

obtain information about chaos characteristics, from a very short train ofpulses, is very important to maintain the constant operation of the whole communications system.

ACKNOWLEDGMENTS

This work was partly supported by CICYT "Comisión Interministerial de Ciencia y Tecnología", grantTIC99-1131 and CAM "Comunidad Autónoma de Madrid", grant 07T/0037/2000.

REFERENCES

1. J.A. Martín-Pereda and A. González-Marcos, "Digital chaos analysis in optical logic structures". **SPIE**, 2612, 170- 180 (1995)
2. A. González-Marcos and J.A. Martín-Pereda, "Digital Chaotic Output from an Opticall -Processing Element", *Optical Engineering* **35**, pp. 525-535, 1996.
3. A. Lempel and J. Ziv, "..", *IEEE Transsac. Inf. Theory* **IT**-22, 75 (1976)
4. A. González-Marcos and J.A. Martín-Pereda, *"Chaotic behaviour evaluation in optical logic gates with fractal concepts"*. Photonic Devices and Algorithms for Computing., *SPIE*, vol.3805, pp. 2-10, (1999)
5. A. González-Marcos and J.A. Martín-Pereda, *"Digital Chaos Synchronization In Optical Networks"*. En "OPTICAL NETWORK DESIGN AND MODELLING II". Eds: G. de Marchis y R.Sabella. Kluwer Academic Publishers, pp. 175-186. 1999
6. A. González-Marcos and J.A. Martín-Pereda, "Optically programmable logic cells as basic units for transmission and synchronization of chaotic signals", paper 4470-09. SPIE's 46th Annual Meeting. San Diego, California, USA (2001)